

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное автономное образовательное учреждение**  
**высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Дисциплина «Системы искусственного интеллекта»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

**Основы языка Python**

**Выполнил (а):**

**Сыроватская Вероника Вадимовна**

**(ФИО)**

**студентка 3 курса 2 группы**

**специальность ППЖ-б-о-20-1**

**очной формы обучения**

**Проверил:**

**Кандидат технических наук**

**Доцент кафедры**

**инфокоммуникаций**

**Воронкин Роман Александрович**

The screenshot shows the Visual Studio Code editor with the file `user.py` open. The code in the editor is as follows:

```
1 nameUser=input("Как вас зовут?")
2 ageUser=input("Сколько вам лет?")
3 cityUser=input("В каком городе вы живете?")
4 print(("Вас зовут {0}, Вам {1} лет и вы живете в городе {2}").format(nameUser, ageUser, cityUser))
5
```

The Run console at the bottom shows the execution of the script:

```
C:\Users\nikas\lab2.1\venv\Scripts\python.exe C:\Users\nikas\lab2.1\user.py
Как вас зовут? Никита
Сколько вам лет? 19
В каком городе вы живете? Ставрополь
Вас зовут Никита, Вам 19 лет и вы живете в городе Ставрополь
Process finished with exit code 0
```

The screenshot shows the Visual Studio Code editor with the file `numbers.py` open. The code in the editor is as follows:

```
1 a=input("Придумайте четыре любых числа и введите первое из них")
2 b=input("Введите второе число")
3 c=input("Введите третье число")
4 d=input("Введите четвертое число")
5 x= int(a) + int(b)
6 y= int(c) + int(d)
7 z= x/y
8 answer= z
9 print("Результат деления суммы первых двух чисел на сумму двух последних равен %.2f" % (answer))
```

The Run console at the bottom shows the execution of the script:

```
C:\Users\nikas\lab2.1\venv\Scripts\python.exe C:\Users\nikas\lab2.1\numbers.py
Придумайте четыре любых числа и введите первое из них 1
Введите второе число 2
Введите третье число 3
Введите четвертое число 4
Результат деления суммы первых двух чисел на сумму двух последних равен 1.42
Process finished with exit code 0
```

The screenshot shows the Visual Studio Code editor with the file `arithmetic.py` open. The code in the editor is:

```
1 answerUser = input("Решите пример: 4*100-54")
2 print("Ваш ответ {}, правильный ответ - 346".format(answerUser))
```

The Run and Debug console shows the following output:

```
C:\Users\nikas\lab2.1\venv\Scripts\python.exe C:\Users\nikas\lab2.1\arithmetic.py
Решите пример: 4*100-54
Ваш ответ 6, правильный ответ - 346
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is 6:1 CRLF UTF-8 with 4 spaces, using Python 3.10 (lab2.1) on the main branch.

The screenshot shows the Visual Studio Code editor with the file `numbers.py` open. The code in the editor is:

```
1 a=input("Придумайте четыре любых числа и введите первое из них")
2 b=input("Введите второе число")
3 c=input("Введите третье число")
4 d=input("Введите четвертое число")
5 x= int(a) + int(b)
6 y= int(c) + int(d)
7 z= x/y
8 answer= z
9 print("Результат деления суммы первых двух чисел на сумму двух последних равен %.2f" % (answer))
```

The Run and Debug console shows the following output:

```
C:\Users\nikas\lab2.1\venv\Scripts\python.exe C:\Users\nikas\lab2.1\individual.py
Input side A
Input side B
Input side C
The area of the triangle is 0.0
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is 9:97 CRLF UTF-8 with 4 spaces, using Python 3.10 (lab2.1) on the main branch.

```

Git CMD
C:\Users\nikas>cd lab2.1
C:\Users\nikas\lab2.1>git add /
fatal: /: '/' is outside repository at 'C:/Users/nikas/lab2.1'

C:\Users\nikas\lab2.1>git add .
warning: in the working copy of '.idea/inspectionProfiles/profiles_settings.xml', LF will be replaced by CRLF the next time Git touches it

C:\Users\nikas\lab2.1>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .idea/.gitignore
    new file:   .idea/inspectionProfiles/profiles_settings.xml
    new file:   .idea/lab2.1.iml
    new file:   .idea/misc.xml
    new file:   .idea/modules.xml
    new file:   .idea/vcs.xml
    new file:   arithmetic.py
    new file:   individual.py
    new file:   numbers.py
    new file:   user.py

C:\Users\nikas\lab2.1>git commit -m "Файл добавлен"
[main 51ddd7c] Файл добавлен

```

```

Git CMD
C:\Users\nikas\lab2.1>git commit -m "Файл добавлен"
[main 51ddd7c] Файл добавлен
10 files changed, 60 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/lab2.1.iml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 arithmetic.py
create mode 100644 individual.py
create mode 100644 numbers.py
create mode 100644 user.py

C:\Users\nikas\lab2.1>git pull
Already up to date.

C:\Users\nikas\lab2.1>git puch
git: 'puch' is not a git command. See 'git --help'.

The most similar command is
  push

C:\Users\nikas\lab2.1>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

```

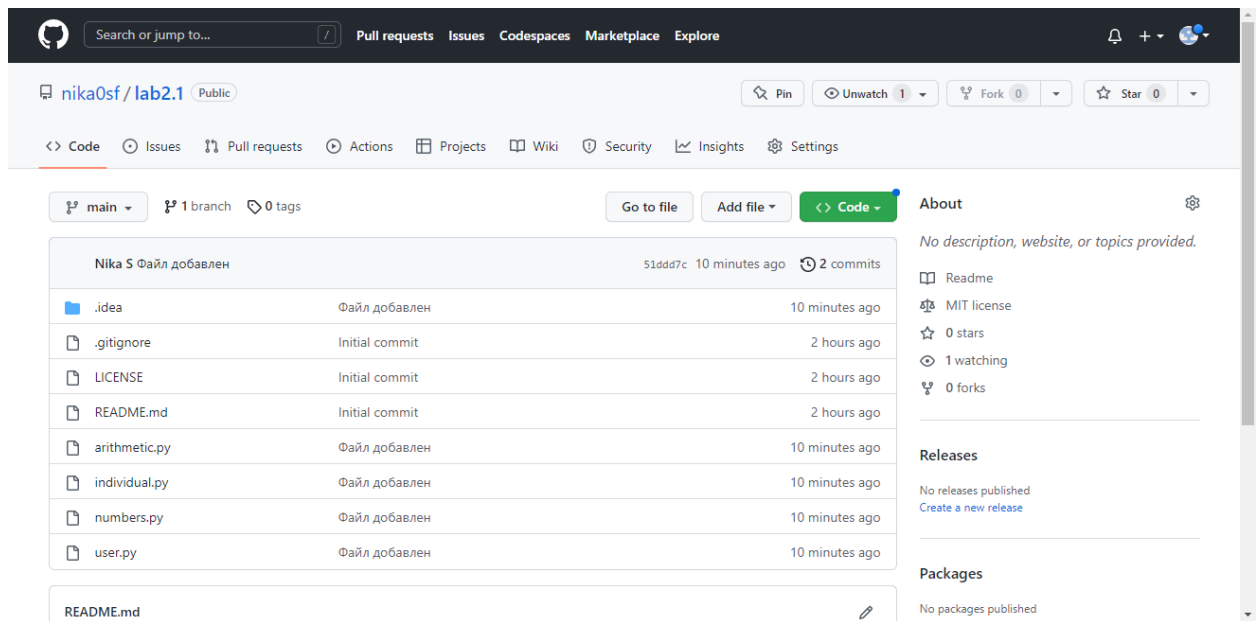
```

nothing to commit, working tree clean

C:\Users\nikas\lab2.1>git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 2.35 KiB | 218.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nika0sf/lab2.1.git
   afaa877..51ddd7c  main -> main

C:\Users\nikas\lab2.1>

```



1. Опишите основные этапы установки Python в Windows и Linux. Windows: • **Запустить скачанный установочный файл** • **Выбрать способ установки** • **Отметить необходимые опции установки** • **Выбрать место установки** Linux: Зачастую интерпретатор Python уже входит в состав дистрибутива. Проверить это можно набрав в терминале `$ python` или `$ python3`. Для установки же их репозитория в Ubuntu необходимо воспользоваться командой `$ sudo apt-get install python3`
2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта? **Anaconda это дистрибутив языков программирования Python, а Python – язык программирования общего назначения высокого уровня.**
3. Как осуществить проверку работоспособности пакета Anaconda? **Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы Пуск Anaconda3 (64-bit) Anaconda Prompt. В появившейся командной строке необходимо ввести `> jupyter notebook` В результате чего отобразиться процесс загрузки веб-среды Jupyter Notebook**
4. Как задать используемый интерпретатор языка Python в IDE PyCharm? **Используемый интерпретатор можно всегда изменить в настройках проекта выбором соответствующего раздела или сочетанием клавиш `Ctrl + Alt + S`**
5. Как осуществить запуск программы с помощью IDE PyCharm?
6. В чем суть интерактивного и пакетного режимов работы Python? **Пакетный режим выполняет готовую программу, интерактивный режим позволяет проверять идеи и варианты решения задач**
7. Почему язык программирования Python называется языком динамической типизации? **В языке Python одна и та же переменная в разное время может ссылаться на данные разного типа**
8. Какие существуют основные типы в языке программирования Python? **В Python есть шесть основных типов данных — числа, строки, списки, кортежи, множества и словари.**

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?
10. Как получить список ключевых слов в Python? **Получить полный список ключевых слов можно используя справочную утилиту интерпретатора Python \$ python3.7 help() help> keywords**
11. Каково назначение функций `id()` и `type()`? **`id()` используется для того, чтобы посмотреть на объект с каким идентификатором ссылается переменная `type()` используется для определения типа переменной**
12. Что такое изменяемые и неизменяемые типы в Python. **К неизменяемым (immutable) типам относятся: целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set). К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict).**
13. Чем отличаются операции деления и целочисленного деления? **При целочисленном делении ответ округляется до наименьшего целого числа, при операции деления выводится неокругленный ответ**
14. Какие имеются средства в языке Python для работы с комплексными числами? **Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде  $a + bj$ .**
15. Каково назначение и основные функции библиотеки (модуля) `math`? **По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`. Библиотека (модуль) `math` содержит большое количество часто используемых математических функций. Например: возвращение ближайшего целого числа, большего чем  $x$ ; возвращение абсолютного значения числа; вычисление факториала  $x$ ; возвращение ближайшего целого числа, меньшего чем  $x$ ; вычисляет  $e^x$ ; логарифмы по основанию 2, 10,  $e$  и любому другому; вычисление значения  $x$  в степени  $y$  и многие другие. Модуль `cmath` работает с комплексными числами, выполняя тригонометрические, гиперболические, логарифмические, классификационные и другие функции.**
16. Каково назначение именованных параметров `sep` и `end` в функции `print()`? **`sep` используется для указания отличного от пробела разделителя строк. Параметр `end` позволяет указывать что делать после вывода строки.**
17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? **Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python. Метод `format()` используется для форматирования строк вывода. Также, есть метод Си-стиля.**